PERFORMANCE EVALUATION OF CONFIGURABLE VITERBI DECODER FOR WIRELESS NETWORK

ZUBER PATEL

ASSISTANT PROFESSOR, S V NATIONAL INSTITUTE OF TECHNOLOGY, SURAT, INDIA.
E-mail: zmp@eced.svnit.ac.in

Abstract - Forward Error Correction (FEC) coding and decoding are key components of wireless communication system. Convolution error correcting codes are very efficient for correction of random error introduced by random noise. The convolutional decoding is mostly done using Viterbi algorithm which is based on predicting sequence of states to get decoded bits. The error correcting capability of Viterbi decoder depends on number of factors such as constrain length, code rate, trace back depth and soft/hard decision. We investigate the effect of constraint length and trace back depth on the performance of Viterbi decoder. The simulation results are analyzed in terms of bit error rate (BER). We then set of specifications for designing decoder that guarantee 10^{-5} at SNR of 4dB.

Keywords - BER, Convolutional codes, Trellis diagram, Viterbi decoding

I. INTRODUCTION

In last few decades, there was enormous growth of data traffic over wireless communication network through mobile and computer communications. The carrier signals that carry information bits undergo changes over the channel. Noise and interference are introduced by channel during signal transmission. This may distort signal of interests and eventually bit error may result. Overcoming bit errors to increase capacity of wireless broadband network in real time is a critical issue. FEC schemes reduce bit errors through encoding and decoding process. FEC encoder adds some carefully designed redundant bits before transmission and FEC decoder estimates transmitted information bits by decoding.

The Viterbi decoding algorithm is a decoding process for convolutional codes for a discrete memory-less channel. For the purpose of error recovery, the encoder adds redundant information to the original information i, and the output t is transmitted through a channel. Input at receiver end (r) is the information with redundancy and possibly, noise. The receiver tries to extract the original information through a decoding algorithm and generates an estimate (e). A decoding algorithm that maximizes the probability p(r|e) is a maximum likelihood (ML) algorithm [1]. An algorithm which maximizes the p(e|r) through the proper selection of the estimate (e) is called a maximum a posteriori (MAP) algorithm [1]. The two algorithms have identical results when the source information i has a uniform distribution. Since the received signal is analog, it can be quantized into several levels. If the received signal is converted into two levels, either zero or one, it is called hard decision. If the input signal is quantized and processed for more than two levels, it is called soft decision. The soft decision captures more information in the input signal consequently performing better than the hard decision at the cost of a higher complexity. In this paper, the ML algorithm with the hard decision output has been used for simulation.

Communication technologies such as Digital Subscriber Line (DSL), Wireless LAN (WLAN), 3G/4G cellular, satellite communication etc. require variations of convolutional coding with differing coding performance at differing data rates and therefore require differing decoding performance, usually using Viterbi decoding. Therefore, there is a need for evaluating convolutional decoder [2] performance using different values of code rate, constraint length [3] and trace back depth. Appropriate values of parameters are selected for target application which guarantees performance and efficient hardware realization. We target WLAN for applying our design. For successful working, decoder should exhibit BER performance of 10^{-5} at E_b/N_o of 4dB.

II. VITERBI DECODING

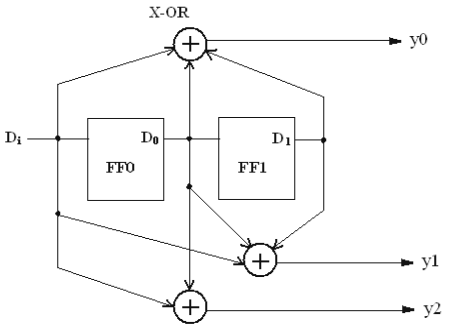

Figure 1: Convolutional encoder with K=3 and rate 1/3

Modern day digital communication system widely uses convolutional codes for error correction. Convolutional codes differ significantly from block codes in structural form and have much more powerful error correcting capability. Convolutional coding can be applied to a continuous input stream (which cannot be done with block codes), as well as blocks of data. Convolutional codes are usually characterized by two parameters and the patterns of n modulo-2 adders (XOR gates). The two parameters are the code rate and constraint length. The code rate, kln, is expressed as a ratio of the number of bits into the convolutional encoder (k) to the number of channel symbols output by the convolutional encoder (n) in a given encoder cycle. The constraint length K denotes the "length", i.e. how many bits are available to feed n modulo-2 adders. In fact, a convolutional encoder can be viewed as a finite state machine having 2^{K-1} states. Its output is function of presents state and present input. The trellis is a time indexed version of the state diagram. Each node corresponds to a state at a given time index, and each branch corresponds.

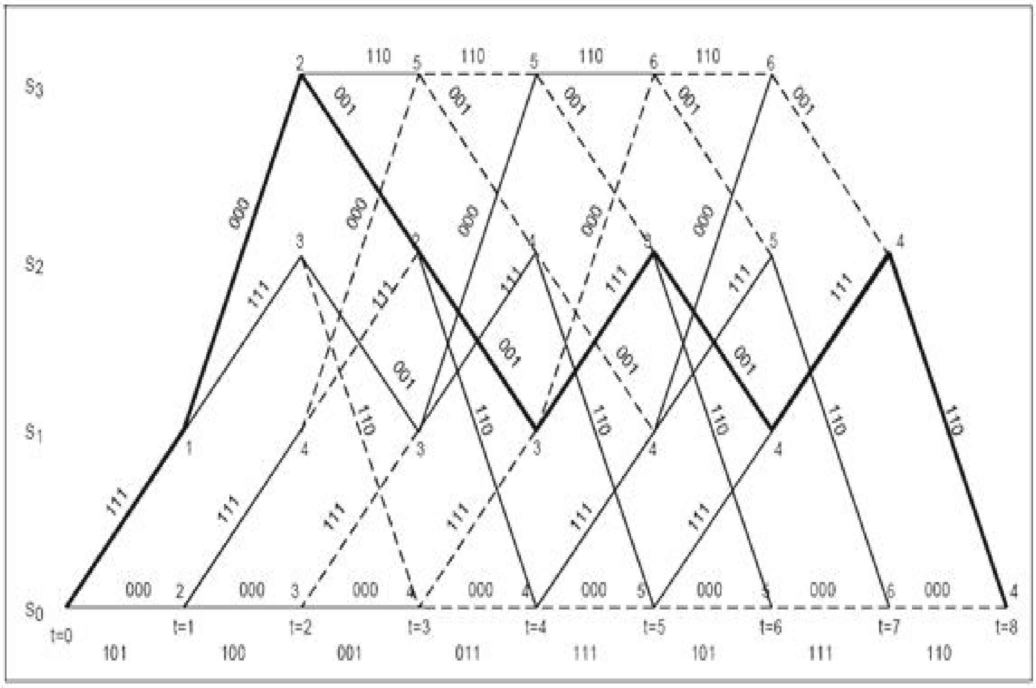

Figure 2: Trellis diagram for encoder of Fig. 1

The Viterbi decoding based on the **ML** algorithm and the hard decision is illustrated in Fig. 2. The trellis in the figure corresponds to the convolutional encoder (Fig. 1) with constraint length 3 and code rate 1/3. The received code symbols are shown at the bottom of the trellis. The encoder encodes an input sequence (11010100) and generates the code word (111,000,001,001,111,001,111,110). This code word is transmitted over a noisy channel, and (101,100,001,011,111,101,111,110) is received at the other end. A branch metric is the Hamming or Euclidean distance between the estimate and the received code symbol. The branch metrics accumulated along a path form a path metric. A path metric at a state, often referred as state metric, is the cumulative sum of branch metrics for the path from the initial state to the given state. After the trellis grows to its maximal size, there are two incoming branches for each node. Between two branches, the branch with a smaller (in terms of Hamming distance) partial metric survives, and the other one is discarded. After surviving branches at all nodes in the trellis have been identified, there exists a unique path starting and ending at the same initial state in the trellis. The decoder generates an output sequence corresponding to the input sequence for this unique path.

III. RELATED WORK

Many architectural designs of Viterbi decoder have been reported in research literatures which optimize some performance metrics by trading with others. The key design specifications are silicon area, speed, latency, power consumption and flexibility. Convolution encoding and decoding is defined by constraint length, coding rate and generator polynomial. Generally, higher performance can be achieved using high constraint length but it increases complexity exponentially. Coding rate can be improved by puncturing technique at the expense of some performance degradation.

Architecture of Viterbi decoder presented in [4] dynamically reconfigures constraint length and trace back depth to achieve area effieceint add-compare-select (ACS) architecture. In [5], authors propose a parallel block-based Viterbi decoder on the graphic processing unit (GPU) platform for the decoding of convolutional codes. It divides received data stream into a series of parallel blocks for concurrent decoding to enhance speed. The work proposed by [6] introduces the pointer implementation for the Register Exchange (RE) method. A pointer is assigned to each row of memory in the survivor memory unit. The content of the pointer which points to one row of memory is altered to point to another row of memory, instead of copying the contents of the first row to the second. This architecture reduces power consumption significantly with little degradation in performance.

The work of [7] implements a reconfigurable adaptive Viterbi decoder for GPRS, EDGE and WiMAX technologies. It dynamically configures constraint length and rate depending on the type of wireless technology. Authors in [8] propose and describe Viterbit decoder for Internet of Things (IoT) application. Simulations are carried out considering 802.11ah network and the system performance is evaluated. It claims to achieve the best BER and PER performance without any hardware trade-off.

IV. PERFORMANCE EVALUATION AND DISCUSSION

The convolutional encoder and Viterbi decoder are modeled in MATLAB software [9] as shown in Fig. 3 along with other required modules. This simulation set up starts with random binary data generator to generate sufficient data for useful BER computations. A binary convolutional encoder of code rate ½ with generator polynomial of [171, 133] octal is used to encode data. The encoded data are modulated with binary phase shift keying (BPSK) and sent to AWGN channel module. Complementary operations are carried out on receiving section. BPSK demodulator produces soft decision using log-likelihood ratio. These soft outputs are 3-bit quantized and passed onto the decoder. Finally, decoded bits and original information bits are sent to error rate calculation module to compute BER. We undertake two experiments to evaluate performance of Viterbi decoder.

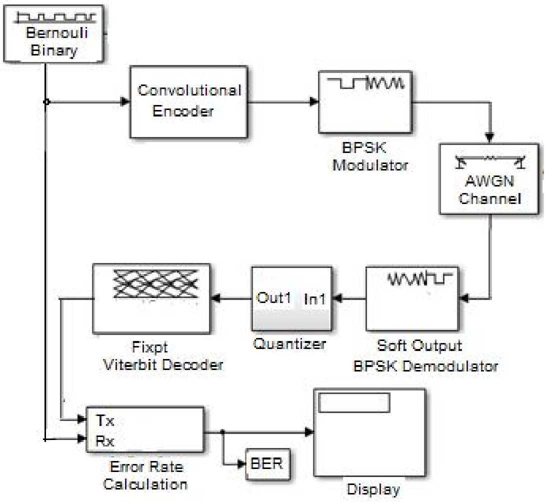

Figure 3: Simulation set up using MATLAB simulink

(a)Experiment 1

In this experiment, we vary the constraint length K of convolutional encoder and analyze its effect on BER performance. The simulation is carried out on an input sequence of 100 million bits iterated over E_b/N_o in the range from 0 to 6dB and 0.5 line spacing in other to obtain a good performance curve. A set of curves have been obtained (as shown in Fig. 4) after BER calculation for the cases of K=6, 7 and 8. It is observed that BER for each constraint length decreases exponentially with the increase in E_b/N_o. To meet the requirement of BER of 10^{-5} at 4dB, two possible values of K are 7 and 8. Since convolution decoding complexity increases exponentially with K, we choose K=7 for the realization of area and power efficient Viterbi decoder.

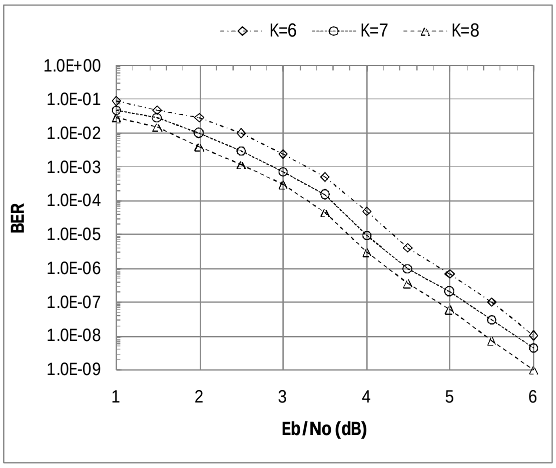

Figure 4: BER plot with different constraint length

(b) Experiment 2

In this experiment, we vary trace back depth D (defined as number of trellis stages that need to be traced in backward direction before decoding sequence of bits) of decoder and analyze its effect on BER performance. As concluded from experiment 1, constraint length K=7 used for this case. Set of curves in plot of Fig. 5 are obtained with D=42, 56 and 70. Clearly, BER performance of Viterbi decoder improves with increase in trace back depth D from 42 to 56. However, D values larger than 56 only exhibits marginal improvement in BER. Thus, for our design we chose D=56.

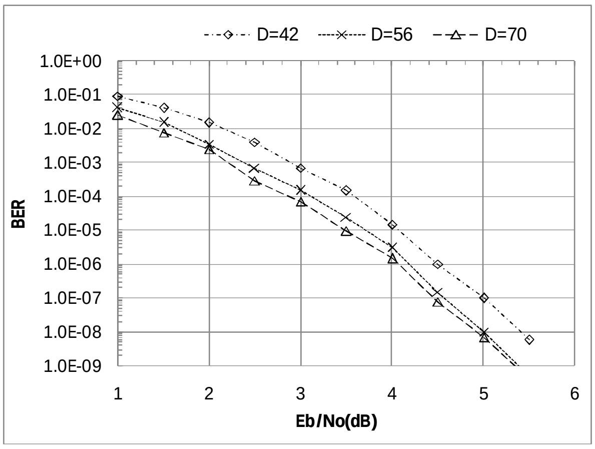

Figure 5: BER plot with different trace back depth

CONCLUSION

The paper discusses convolutional codec with configurable constraint length and trace back depth. Iterative simulations are undertaken in MATLAB to decide specification of Viterbi decoder for WLAN application. BER plots are analyzed to determine practical values of constraint length K and trace back depth D to be used for achieving target BER of 10^{-5}. We simulated Viterbi decoder in MATLAB iteratively by varying K and D. Plots reveal that k=7 and D=56 is sufficient to decode data stream for rate ½ without much noticeable performance degradation.

REFERENCES

[1] S. B. Wicker, Error Control Systems for Digital Communication and Storage - Prentice Hall,1995.

[2] Ekwe A.O., Iroegbu C. and Okoro K.C., “Performance Evaluation of Convolutional Encoders with Different Constraint Rates,” International journal for research in applied science and engineering technology (IJRASET), Vol.2, No.VIII, Aug. 2014.

[3] A. Kulkarni, D. Mantri, N. R. Prasad and R. Prasad, “Convolutional Encoder and Viterbi Decoder Using SOPC For Variable Constraint Length,” in Proc. IEEE Int. Conf. on Advance Computing Conference (IACC), DOI: 10.1109/IAdCC.2013.6514476, 2013.

[4] M. Benaissa and Y. Zhu, “A Novel High-Speed Configurable Viterbi Decoder for Broadband Access,” EURASIP Journal on Applied Signal Processing, Vol.13, pp. 1317–1327, 2003.

[5] H. Peng, R. Liu, Y. Hou and L. Zhao, “A Gb/s Parallel Block-based Viterbi Decoder for Convolutional Codes on GPU,” 8th Int.Conf. on Wireless Communications & Signal Processing (WCSP), DOI: 10.1109/WCSP.2016.7752638, 2016.

[6] Dalia A. El-Dib and M. I. Elmasry, “Memoryless Viterbi Decoder,” IEEE Trans. on Circuits and Systems—II: Express Briefs, Vol. 52, No. 12, December 2005.

[7] M.F.N. Batcha and A. Z. Sha’ameri, “Configurable Adaptive ViterbiDecoder for GPRS, EDGE and WiMAX,” in Proc. of IEEE Int. Conf. on Telecommunications and Malaysia International Conference on Communications (ICT-MICC 2007), Malaysia, DOI: 10.1109/ICTMICC.2007.4448640, May 2007

[8] T. H. Tran, H. Kato, S. T. Yamazaki and Y. Nakashima, “Performance Evaluation of 802.11ah Viterbi Decoder for IoT Applications.” In Proc. IEEE International Conference on Advanced Technologies for Communications (ATC), DOI: 10.1109/ATC.2015.7388343, 2015.

[9] Viterbi Decoder, MATLAB Communication System Tool Box- Available at: https://in.mathworks.com/help/comm/ref/viterbidecoder.html.

★★★